**3-3**

УДК 519.64:517.443:519.254

***В.К. Задирака***
Институт кибернетики имени В.М. Глушкова НАН Украины
Проспект Академика Глушкова, 40, Киев, 03680 МСП, Украина

# Компьютерная технология интегрирования быстроосциллирующих функций с заданными значениями характеристик качества по точности и быстродействию

***V.K. Zadiraka***
*Glushkov Institute of Cybernetic of NAS of Ukraine*
*40 Glushkova ave., Kyiv, Ukraine, 03187*

# *Computer Technology of Integration of Quickly Oscillations Functions with the Set Values Characteristics of Quality in Terms of Exactness and Fast-acting*

***В.К. Задірака***
Інститут кібернетики імені В.М. Глушкова НАН України, м. Київ, Україна
Проспект Академіка Глушкова, 40, Київ, 03680 МСП, Україна

# Комп'ютерна технологія інтегрування швидкоосцилюючих функцій із заданими значеннями характеристик якості за точністю і швидкодією

Рассмотрены вопросы выбора и построения вычислительных ресурсов и способы эффективного их использования при приближенном вычислении интегралов от быстроосциллирующих функций из некоторых классов с заданной точностью за ограниченное процессорное время. Приведена соответствующая компьютерная технология.
**Ключевые слова:** компьютерная технология, оценка погрешности, оптимальные алгоритмы.

The problems of calculation resources choice and construction and ways to effectively use them in the approximate calculation of integrals from quickly oscillations functions of certain classes with defined exactness for the limited processor time are considered. The appropriate computer technology is proposed.
**Key words:** computer technology, estimation of error, optimal algorithms.

Розглянуто питання вибору та побудови обчислювальних ресурсів та способу ефективного їх використання при наближеному обчисленні інтегралів від швидко осцилюючих функцій з деяких класів із заданою точністю за обмежений процесорний час. Наведена відповідна комп'ютерна технологія.
**Ключові слова:** комп'ютерна технологія, оцінка похибки, оптимальні алгоритми.

**Цель работы.** Поскольку «легкие» задачи уже перерешали и остались «трудные» задачи, которые невозможно решить при помощи штатного материального обеспечения с требуемым качеством, необходимо привлекать для ее решения резервы оптимизации вычислений [1]. Как их использовать и в каком порядке, подсказывает компьютерная технология решения задач с заданными значениями характеристик качества.

**Постановка задачи.** Ниже будут приведены элементы компьютерной технологии построения вычислительных алгоритмов (в.а.) с заданными значениями характеристик качества для решения задачи интегрирования быстроосциллирующих функций вида:

$$I(\omega)=\int_a^b f(x)\begin{Bmatrix} e^{-i\omega x} \\ \sin\omega x \\ \cos\omega x \end{Bmatrix}dx,$$

и

$$W_f(s,f)=S^{-1/2}\int_{-\infty}^{\infty} f(x)\cdot\psi\left(\frac{x-t}{S}\right)dx, \qquad (1)$$

где $f(x)\in F$ ($F$ – некоторый класс функций); $\omega$ – произвольное вещественное число ($|\omega|\geq 2\pi(b-a)$). Информация о $f(x)$ задана $N$ значениями в узловых точках $\{x_i\}_0^{N-1}$ из области ее определения: $\{f_i\}_0^{N-1}=\{f(x_i)\}_0^{N-1}$, $\varepsilon_i$ характеризует точность задания $f(x_i)$: $|\tilde{f}_i-f_i|\leq\varepsilon_i,\ i=\overline{0,N-1}$; $\psi(x)$ – вейвлет-фукнция [2], которая колеблется около оси *x*, удовлетворяет условию $\int_{-\infty}^{\infty}\psi(x)dx=0$ и быстро стремится к нулю на бесконечности.

Пусть $c(Y)$ – модель компьютера, $c(Y)\in C(Y)$ ($C(Y)$ – класс моделей компьютеров).

Общая ситуация построения вычисления интегралов (1) с точностью $\varepsilon$, ($\varepsilon>0$) при ограниченных вычислительных ресурсах описывается условиями

$$E(I,X,Y)\leq\varepsilon, \qquad (2)$$

$$T(I,X,Y)\leq T_0(\varepsilon), \qquad (3)$$

$$M(I,X,Y)\leq M_0(\varepsilon), \qquad (4)$$

где $\varepsilon,T_0,M_0$ – заданные числа; $E(I,X,Y)$ – полная погрешность приближенного вычисления интегралов (1), которая является суммой трех составляющих: $E_н(\cdot)$ – неустранимой погрешности за счет неточности входных данных; $E_M(\cdot)$ – погрешности метода; $E_\tau(\cdot)$ – погрешности округлений; $X,Y$ – векторы параметров, которые характеризуют соответственно в.а. и компьютеры из классов $A$ и $C$; $T(\varepsilon,I,X,Y)$ и $M(\varepsilon,I,X,Y)$ – процессорное время и память компьютера, необходимые для приближенного вычисления интеграла.

Приближенное значение интеграла (1), для которого выполняется условие (2), называется $\varepsilon$-решением, $A(\varepsilon,X,Y)$ – множество в.а. построения $\varepsilon$-решения в данной модели вычислений.

Вычислительный алгоритм, который удовлетворяет условиям (2), (3), будем называть *T*-эффективным.

Далее будем предполагать (если не допускается другое), что память *M* может быть расширена до необходимого объема, то есть ограничение (4) можно снять, но, скорее всего, за счет увеличения характеристики *T* (процессорного времени). Это может быть

осуществлено, например, путем наращивания части «медленной» дисковой памяти в общей структуре компьютерной памяти. Учитывая тот факт, что при $\varepsilon \to 0$, $M_0 \to \infty$ (например, когда идет речь о погрешности округлений или погрешности метода), будем считать, что $\varepsilon \geq \varepsilon_0 > 0$, где $\varepsilon_0$ – задано.

На практике используют некоторые оценки полной погрешности в.а., ее составляющих и процессорного времени. При этом различают оценки априорные и апостериорные, мажорантные и асимптотические, детерминированные и вероятностные. Возможность и целесообразность использования указанных оценок и способов их получения зависит от типа, структуры и точности априорных данных задачи и в.а. от того, с какой целью вычисляется оценка, а также от вычислительных ресурсов.

Мажорантные априорные оценки гарантируют верхнюю грань оцениваемой величины и выражаются через известные величины, вычисление их не требуют значительных вычислительных затрат, но значения оценок часто очень завышены, поэтому выводы на их основании касательно возможности вычисления интеграла при условиях (2), (3) могут быть ошибочными.

Асимптотические оценки аппроксимируют величину, которая оценивается. Варьированием параметра может быть достигнута желаемая близость оценки к оцениваемой величине, но вычисление таких оценок связано со значительными вычислительными затратами. Эти оценки, как правило, апостериорные.

В алгоритмическом обеспечении вычисления интеграла (1) при соблюдении условий (2), (3), в соответствии с указанными характеристиками оценок, нужно предусмотреть возможности вычисления разных видов оценок характеристик $E(E_н, E_M, E_\tau)$ [3]. При ослабленных ограничениях (2), (3) достаточными оценками могут быть менее точные и менее сложные (в вычислительном плане). При усиленных ограничениях (2), (3) используются асимптотические (апостериорные) оценки. Например, условие (3) может привести к высоким требованиям к точности параметров вычислительного процесса, которые вычисляются на основе оценок погрешности решения.

Квадратурные формулы вычисления интегралов типа (1) широко используются при решении следующих важных классов прикладных задач: прикладной статистики, анализа сейсмограмм землетрясений, анализа и синтеза речевых сигналов, решения краевых задач для уравнений в частных производных, информационной безопасности и др. Поэтому приближенное вычисление интегралов (1) для $f(x)$ из некоторых классов и анализ точности соответствующих квадратурных и кубатурных формул являются актуальными задачами вычислительной и прикладной математики.

# Компьютерная технология

В случае, если не удается построить в.а. $a \in A(\varepsilon, T_0)$ вычисления $\varepsilon$-решения задачи (1) на классах функций $f \in F, F_N, F_{N,\varepsilon}$ [4], для окончательного вывода о возможности такого решения важно иметь точные оценки снизу (или близкие к ним) приближенного решения и вычислительной сложности задачи. Воспользовавшись этими оценками в.а., можно сделать окончательный вывод: решение задачи с заданными значениями характеристик качества можно построить, или что такое решение построить невозможно и, возможно, необходимо $f(x)$ погрузить в более узкий класс функций (например, $F_N$ или $F_{N,\varepsilon}$) или использовать компьютер другого класса.

Для улучшения качества квадратурных формул вычисления интегралов от быстроосциллирующих функций (1) целесообразно использовать следующие резервы оптимизации вычислений:

1. Резервы уменьшения меры погрешности:

а) за счет уточнения входных данных ($E_н$);

– уточнение класса подынтегральных функций;

– корректировка входной информации;

– повышение точности входной информации.

б) Метода $E_M$ :

– использование оптимальных и близких к ним по точности в.а.;

– оптимизация информационного набора функционалов (например, оптимальная сетка узлов);

– увеличение количества функционалов в информационном наборе;

– использование иного информационного оператора;

– полное использование входной информации для сужения класса подынтегральных функций.

в) Округлений ($E_\tau$):

– использование методов оптимального порядка точности для высокоточных вычислений;

– сужение классов (с целью уменьшения количества функционалов в информационном наборе);

– использование схем вычислений, которые минимизируют скорость накопления погрешности округлений;

– увеличение оптимального порядка точности путем выбора класса информационных операторов;

– выбор и моделирование правила округления.

2. Резервы уменьшения процессорного времени:

– увеличение длины разрядной сетки;

– использование оптимальных и оптимальных по порядку точности квадратурных формул;

– использование уточненной входной информации (как следствие – менее жесткие ограничения на погрешности метода и округлений);

– улучшение качества оценок погрешностей метода и округлений;

– улучшение точности вычисления параметров вычислительного процесса;

– использование алгоритмов быстрых ортогональных преобразований;

– использование «быстрой» арифметики;

– согласование в.а. с архитектурой компьютера;

– распараллеливание вычислений.

Необходимость в вычислении интегралов (1) с необходимыми значениями характеристик качества и в получении информации о возможности обеспечения такого решения возникает, например, в следующих случаях:

– необходима диагностика качества полученного значения интеграла по точности, времени и необходимой памяти компьютера;

– необходимо до нахождения значения интеграла знать о возможности его нахождения с необходимыми ограничениями на $E$, $T$, $M$ ;

– необходимо обеспечить необходимую точность вычисления интеграла (1).

Программа для вычисления значения интеграла может принадлежать к одному из следующих типов:

1) программа нахождения приближенного значения интеграла с заданной точностью. Значения ее управляющих параметров, которые соответствуют необходимой точности $\varepsilon > 0$, вычисляются самой программой;

2) программа, которая вычисляет $\varepsilon$-решение задачи и вычисляет оценки погрешности $E(I,X,Y)$ полученного решения;

3) программа, которая находит приближенное (с точностью $\varepsilon > 0$) значение интеграла (без вычисления оценки погрешности).

Каждая из приведенных программ может вычислять апостериорные оценки $E, T, M$.

Для того, чтобы еще до решения задачи можно было судить о возможном обеспечении ее решения с необходимыми характеристиками качества, необходимо иметь соответствующую программу вычисления априорных оценок ее характеристик $E$, $T$, $M$, оценок наилучших управляющих параметров $X$ (если такие у нее есть), а также выдачи информации о значениях оценок ее характеристик, полученных в результате тестирования.

Технология вычисления интеграла обеспечивается пользователем в диалоге с компьютером с помощью программ решения задачи и вычисления априорных оценок ее характеристик. Она существенно зависит от типа программ, к которому она принадлежит, а также от наличия программы вычисления оценок характеристик.

Так, если программа вычисления интеграла принадлежит к первому типу, то при заданной (из допустимого диапазона) точности вычисления интеграла вычисляются оценки $T$ и $M$. Если они соответствуют заданным ограничениям (3), (4), интеграл может быть вычислен с заданными значениями характеристик качества.

В случае, когда программа принадлежит ко второму или третьему типу, то на основании вычисленных априорных оценок $E,\ T,\ M$ можно судить о возможности выполнения ограничений (2) – (4). Если программа не вычисляет априорную оценку $E$, а только оценки $T$ и $M$, то вычисляются только они. Если ограничения на $T$ и $M$ выполняются, то задачу можно решать выбранной программой. При этом для программы второго типа при помощи вычисленной апостериорной оценки точности можно сделать вывод о выполнении ограничения (2).

О выполнении (для данной программы) ограничений (2) – (4) можно сделать вывод и при помощи результатов ее тестирования [5].

Для обеспечения $\varepsilon$-решения задачи (1) с заданными значениями характеристик качества (2) – (4) необходимо знать, что нужно делать, как это делать и в какой последовательности. Приведем описание последовательности шагов технологической схемы, которая реализует этапы приведенной выше схемы решения задач.

**Шаг 1.** Выбирается интеграл (см. (1)), который необходимо приближенно вычислить. Формулируются требования к $\varepsilon$-решению. Фиксируется класс подынтегральных функций ($F,\ F_N,\ F_{N,\varepsilon}$).

В дальнейшем (при возвращении к шагу 1) может быть рассмотрен более узкий класс подынтегральных функций с целью расширения возможности удовлетворения условиям (2) – (4).

**Шаг 2.** Находится (или улучшается) оценка меры погрешности искомого $\varepsilon$-решения за счет неточности входных данных ($E_н$), используя компьютерную модель вычислений, входные данные и оценки погрешности ее задания.

**Шаг 3.** Для найденной оценки $E_н$ проверяется выполнение условия

$$E_н \le \alpha_i \cdot \varepsilon,\ 0 < \alpha_i \le 1 \text{ (например, } \alpha_i = 1/3\text{)}. \quad (5)$$

Если условие (5) выполняется, перейти на шаг 5, иначе – на шаг 4.

**Шаг 4.** Для обеспечения $\varepsilon$-решения необходимо изменить (если это возможно) требование на точность (увеличить $\varepsilon > 0$) или улучшить (уменьшить $E_н$), используя определенные резервы. Перейти на шаг 2.

В случае, когда $E_н$ больше заданного $\alpha_i \cdot \varepsilon$ и не может быть уменьшено, необходимо изменить постановку задачи, то есть перейти на шаг 1, или прекратить поиск $\varepsilon$-решения в связи с невозможностью его вычисления в данной компьютерной модели вычислений.

**Шаг 5.** Изучить вычислительную ситуацию для решения задачи.

5.1 Для вычисления интеграла неизвестен в.а. и соответственно теоретически не изучены оценки его характеристик. Перейти на шаг 6.

5.2 Известен метод вычисления интеграла, но не построены на его основе в.а. и программы для соответствующего компьютера $c(Y)$ и теоретически не изучены оценки их характеристик ($E, T, M$); перейти на шаг 7.

5.3 Для вычисления интеграла имеются алгоритмы – программы из некоторых известных программных систем (например, Mathematica, Matlab, Mathcad), но они не обеспечены оценками характеристик, за счет использования которых можно было бы сделать вывод о возможности найти $\varepsilon$-решение интеграла, удовлетворяющее условиям (2) – (4); перейти на шаг 13.

5.4 Для вычисления интеграла имеется пакет прикладных программ (см., например, [3], в котором есть в.а. – программа нахождения $\varepsilon$-решения задачи; особенность этого пакета состоит в том, что его в.а. – программа нахождения $\varepsilon$-решения задачи сопровождается оценками характеристик $E, T, M$; если она обеспечивает нахождение $\varepsilon$-решения с заданной (в некотором диапазоне $D/\varepsilon$) точностью $\varepsilon \in D(\varepsilon)$, то перейти на шаг 14, иначе – на шаг 7.

**Шаг 6.** Для нашей задачи не разработан метод ее решения, и его необходимо создать. Перейти на шаг 7.

**Шаг 7.** Для решения задачи вычисления интеграла известны (или разработаны в соответствии с шагом 6 методы (или метод).

Из множества известных методов приближенного вычисления интегралов, используя известные априорные оценки их характеристик (например, оценки погрешности метода, сложности), выбирается «лучший» по точности и быстродействию, который может быть использован для вычисления интеграла. С этой целью используются известные знания о классе задач, которые имеются в компьютерных базах знаний, статьях, обзорах, монографиях (см., например, [4], [6]).

**Шаг 8.** Выбираются параметры компьютера, на котором будет решаться задача, исходя из начального анализа ее сложности и требований к характеристикам качества $\varepsilon$-решения, то есть, доопределяется компьютерная модель $c(Y) \in C(Y)$ путем выбора параметров $Y$: количество процессоров ($k = 1$ или $k > 1$) и их тип, длина машинного слова, количество машинных слов, которые используются при реализации арифметических операций, оценки времени выполнения операций, режим вычислений, ограничения $M_0$ на используемую оперативную память.

**Шаг 9.** На основе разработанного или выбранного метода разрабатывается *T*-эффективный в.а. (см. [7]) вычисления $\varepsilon$-решения задачи и при этом находятся оценки его характеристик при помощи следующей технологической схемы.

9.1 На основании выбранного метода для модели компьютера $c(Y)$ разрабатывается некоторый в.а. (с той или иной детализацией его описания, например, пошаговое

описание, описание на алгоритмическом языке), или разрабатывается новый или улучшается разработанный (при повторных реализациях шага 9.1) вычислительный алгоритм.

9.2 Находится или улучшается известная априорная оценка $M(\varepsilon,I,X,Y)$ необходимой оперативной памяти компьютера, которая будет использоваться в.а. при построении приближенного решения задачи.

9.3 Проверяется выполнение условия (4). Если оно выполняется, то перейти на шаг 9.4; иначе – на шаг 8 (или 9, если алгоритм фиксирован). Это делается с целью выполнения условия (4) за счет модификации разработанного или построения нового в.а., возможно с выбором соответствующего компьютера $c(Y)$.

9.4 Находятся или улучшаются априорные и (или) апостериорные оценки: $E_н$, $E_м$, $E_\tau$ – соответственно за счет неточности входной информации, метода (алгоритма), округлений – при использовании в.а. $a \in A$ и модели компьютера $c(Y)$. (Примеры получения таких оценок в [4], [6].

9.5 На основании оценки $E$ проверяется, выполняется ли условие (2), то есть, может ли быть вычислен интеграл с точностью $\varepsilon$ при заданной информации $I_n$. Если $\varepsilon$-решение не обеспечивается, то выясняются причины этого и обсуждаются способы их устранения (осуществляется анализ, начиная с шага 7 или 9.1 данной технологии). При обеспечении нахождения $\varepsilon$-решения выбранным в.а. перейти на шаг 9.6, в противном случае поиск в.а. прекращается и указывается, почему невозможно обеспечить выполнение условия (2).

9.6 Находится или улучшается априорная оценка процессорного времени $T(\varepsilon,I,X,Y)$ в.а. (Примеры таких оценок см., например, в [7]).

9.7 Используя оценку $T(\varepsilon,I,X,Y)$ в.а., выясняется, обеспечивает ли этот в.а. $\varepsilon$-решение задачи при условии (3). Если $\varepsilon$-решение обеспечивается при этом условии, то тем самым будет установлено, что найденный в.а. принадлежит множеству $A(\varepsilon,T_0)$ ($A(\varepsilon,T_0) \in A(\varepsilon)$ эффективных по быстродействию ($T$-эффективных) в.а., которое зависит от выбора $T_0$. Если же в.а. не является $T$-эффективным, то выясняются возможности обеспечения его $T$-эффективности за счет использования резервов оптимизации в.а. (перейти на шаг 9.1 или 9.6).

## Замечание

Если, используя приведенную технологию, не удается построить в.а. вычисления $\varepsilon$-решения, для окончательного вывода о возможности такого построения важно иметь точные оценки снизу (или близкие к ним) погрешности приближенного решения и вычислительной сложности задачи вычисления интеграла. Используя эти оценки в.а., можно сделать окончательный вывод: $\varepsilon$-решение можно построить или такое решение построить невозможно при данных условиях вычислений и, скорее всего, необходимо использовать более точную исходную информацию, большего объема или использовать другой компьютер.

Если разработан $T$-эффективный в.а. с указанными выше значениями характеристик, то перейти на шаг 10, иначе – на шаг 9.

**Шаг 10.** Разработанный $T$-эффективный в.а. реализуется в в.а.-программе на соответствующем алгоритмическом языке для выбранной модели компьютера.

Разработанная в.а.-программа, которая обеспечивает решение задачи с необходимыми значениями характеристик качества (2) – (4), должна принадлежать к одному из трех указанных выше типов программ.

**Шаг 11.** Осуществляется тестирование разработанной в.а.-программы и построенных для нее оценок характеристик $E$ и $T$ . Тестирование осуществляется на соответствующем наборе тестовых задач и в соответствии с технологией тестирования характеристик качества в.а.-программы (см., например, [5]).

Если результаты тестирования подтверждают тот факт, что разработанная (или выбранная) в.а.-программа может обеспечить построение $\varepsilon$ -решения задачи с необходимыми значениями характеристик качества (2) – (4), то перейти на шаг 15, иначе – на шаг 12.

**Шаг 12.** Проводится анализ в.а.-программы с целью выявления резервов ее улучшения (оптимизации) по тем характеристикам, которые не удовлетворяют заданным условиям; осуществляется оптимизация в.а.-программы и для полученной модификации выполняются действия, указанные на 9-м шаге; если исчерпаны все резервы оптимизации для выбранного в.а. и соответствующей программы, но ее характеристики не удовлетворяют заданным ограничениям (2) – (4), то необходимо продолжить поиск необходимого в.а. в соответствии с предыдущими шагами (то есть, переход на шаг 8 или 9).

**Шаг 13.** Для выбранной программы вычисления интеграла проводится (если есть необходимость) ее тестирование и получение оценок характеристик: $E$ – точности вычисления интеграла, $T$ – времени для вычисления интеграла в соответствии с технологией тестирования (см., например, [5]).

Если выбранная в.а.-программа в соответствии с результатами тестирования может обеспечить построение $\varepsilon$ -решения с необходимыми значениями характеристик качества по точности и быстродействию, то перейти на шаг 15, иначе – на шаг 5.2 или 5.3.

**Шаг 14.** Используя оценки характеристик $E, T, M$ в.а.-программы вычисления интеграла с точностью $\varepsilon$ , проверяется, может ли она обеспечить $\varepsilon$ -решение с заданной (2) точностью $\varepsilon \in D(\varepsilon)$ и при заданном ограничении на компьютерное время (3). Если эта в.а.-программа может обеспечить вычисление $\varepsilon$ -решения задачи, то перейти на шаг 15, иначе – на шаг 5.

**Шаг 15.** Вычисление (разработанной или выбранной в.а.-программой) $\varepsilon$ -решения с заданными значениями характеристик качества (2) – (4).

**Шаг 16.** Интерпретация результатов вычисления.

# Выводы

Предложенная и пошагово описанная информационно-компьютерная технология вычисления интеграла с заданными значениями характеристик качества решения (2) – (4) дает возможность:

– осуществить системный подход как к постановке задачи, так и к ее решению;

– изучить вычислительную ситуацию (см. шаг 5) при вычислении интеграла и выявить все шаги, которые необходимо осуществить, чтобы обеспечить решение задачи с заданными значениями (2) – (4) характеристик качества решения;

– при данной вычислительной ситуации осуществить разработку соответствующей в.а.-программы для решения задачи;

– использовать соответствующую программу из известных программных систем (например, Mathematica, Matlab, Mathcad), осуществляя (по необходимости) тестирование для выявления той программы, которая обеспечивает решение задачи с заданными значениями характеристик качества решения;

– использовать предложенную информационно-компьютерную технологию при решении задачи с повышенными требованиями к значениям (2) – (4) характеристик качества ее решения.

# Литература

1. Задирака В.К. Об использовании резервов оптимизации вычислений для улучшения качества вычисления интегралов от быстроосциллирующих функций / В.К. Задирака, С.С. Мельникова, Л.В. Луц // Кибернетика и системный анализ. – 2010. – № 6. – С. 51-64.
2. Дьяконов В.П. Вейвлеты. От теории к практике / Дьяконов В.П. – М. : Солон-Р, 2002. – 429 с.
3. Комплекс программ ПОМ-1 для решения некоторых классов задач вычислительной математики [Иванов В.В., Бабич М.Д., Березовский А.И. и др.] // Управляющие системы и машины. – 1986. – № 2. – С. 105-109.
4. Задирака В.К. Теория вычисления преобразования Фурье / Задирака В.К. – Киев : Наук. думка, 1983. – 215 с.
5. Бабич М.Д. Вычислительный эксперимент в проблеме оптимизации вычислений / М.Д. Бабич, В.К. Задирака, И.В. Сергиенко // Кибернетика и системный анализ. – 1999. – Ч. 1, № 1. – С. 51-63; 1999. – Ч. 2, № 2. – С. 59-79.
6. Оптимальні алгоритми обчислення інтегралів від швидкоосцилюючих функцій та їх застосування. – Т. 1 : Алгоритми. – 448 с.; Т. 2 : Застосування. – 348 с. / [І.В. Сергієчко, В.К. Задірака, О.М. Литвин, С.С. Мельникова, О.П. Нечуйвітер]. – Київ : Наук. думка, 2011.
7. *Т*-ефективні алгоритми наближеного розв'язання задач обчислювальної та прикладної математики / [Задірака В.К., Бабич М.Д., Березовський А.И., Бесараб П.Н., Людвиченко В.О.]. – Тернопіль : Збруч, 2003. – 261с.

# Literatura

1. Zadyraka VK On the use of reserves calculations optimization to improve the evaluation of integrals of rapidly oscillating functions / VK Zadyraka, SS Melnikova, L. Lutz / / Cybernetics and Systems Analysis tion. – 2010. - № 6. – S. 51-64.
2. Deacons VP Wavelets. From theory to practice / Deacons V.P. – Moscow : Solon P, 2002. – 429 p.
3. Complex programs POM-1 solutions for some classes of problems in computational mathematics [V.V. Ivanov, M.D. Babich, A.I. Berezovsky and others] // Control systems and machines. – 1986. – № 2. – S. 105-109.
4. Zadyraka VK The theory of Fourier transformation / V.K. Zadyraka. – Kiev : Science. Dumka, 1983. – 215 p.
5. Babich M.D. Computer experiment in the problem of improving computing / M.D. Babich, V.K. Zadyraka, I.V. Sergienko // Cybernetics and Systems Analysis. – 1999. – Part 1, № 1. – S. 51-63, 1999. – Part 2, № 2. – S. 59-79.
6. Optimalni algorithmic obchislennya integraliv od shvidkoostsilyuyuchih funktsiy that ïh zastosuvannya. - Volume 1: Algorithms. - 448 p., T. 2 Zastosuvannya. - 348. / [I.V. Sergiєchko, V.K. Zadiraka, O.M. Litvin, SS Melnikova, OP Nechuyviter]. – Mumbai : Science. Dumka, 2011.
7. T-efektivni nablizhenogo rozv'yazannya algorithmic tasks that obchislyuvalnoï prikladnoï Mathematics / [Zadiraka V.K. Babich M.D., A. Berezovsky, Besarab P.N., Lyudvichenko V.O.]. – Ternopil : Zbruch, 2003. – 261 s.

***RESUME***

***V.K. Zadiraka***

## *Computer Technology of Integration of Quickly Oscillations Functions with the Set Values Characteristics of Quality in Terms of Exactness and Fast-acting*

The problems of calculation resources choice and construction and ways to effectively use them in the approximate calculation of integrals from quickly oscillations functions of certain classes with defined exactness for the limited processor time are considered. The appropriate computer technology is proposed.

*Статья поступила в редакцию 19.04.2013.*